MIE 1624 Assignment 2

Ziruo Song (Zorina)

Nov 9, 2022

Background and Dataset

The dataset was derived from the “2022 Kaggle Machine Learning & Data Science Survey” dataset, and the answers of Q29, current yearly compensation in US dollars, were grouped by 15 salary buckets in order and encoded from 0 to 14. The purpose of this exploration is to train, validate, and tune multi-class ordinary classification models that can classify what a survey respondent’s current yearly compensation bucket is.

Part 1. Data cleaning

The original cleaned dataset consisted of both single and multi-selecting questions in categoric data, and most of the two types contains loads of missing values. Firstly, the duration of filling the survey was dropped for the further exploration. The options of the multi-choice questions were labeled with 1 if being selected and 0 otherwise. Then the single-selecting questions with more than half missing values were also dropped completely as there would be a misleading effect if labels were given improperly, like Q22 which asked about the ML model hubs used most often personally. Q9 about whether any academic research was published were replaced with ‘Unknown’ if being missing which might be attributed to the ambiguous definition of the academic research. The others asking about the preference or related with calculations, like spending, or years might be hard to decide for the participants, so the missing values of the features containing these properties were filled with the mode. Secondly, convert the categorical data of single-selecting features in numerical ways. The features with ordinal records, like the years of using machine learning methods, were encoded with numbers in ascending order, while the nominal features without order property was replaced with dummy variables that each option took values 1 or 0. As a result, 376 input features and encoded salary was shown numerically without any missing values for the further exploration.

Part 2. Exploratory data analysis and feature selection

By selecting and transforming the most relevant variables from the raw data, feature engineering is a useful tool to create a predictive model in machine learning. By *Figure 1*, the output feature was imbalanced, and using the whole 376 features may cause overfitting, so feature selection could allow overfitting reducing for improving accuracy, or saving the training time on unseen data. Here, resulting in keeping only the most significant variable but forcing the coefficients of some less contributive variables to be exactly zero, lasso regularization model was performed on 70% training data with grid search to get the best hyper-parameter. Then 15 input features were remained according to their importance on affecting salary bucket, shown in *Figure 2*. The correlation of each input feature with the encoded salary were performed and visualized in *Figure 2*. As there was no correlation bigger than 0.6 among any two of the input features in *Figure 3*, we assume there was no obvious dependence. We could see the top features de-

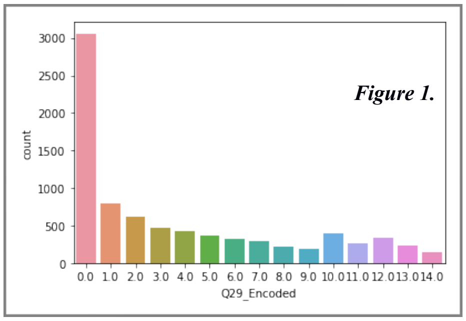


Figure 1.

termine the salary bucket was whether the individual resides in America or not. More discussion about the feature importance was talked in the comparison later

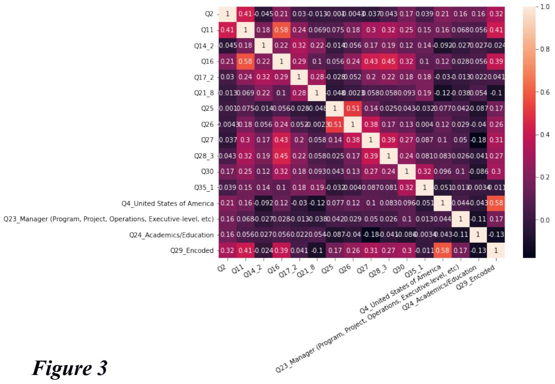


Figure 3

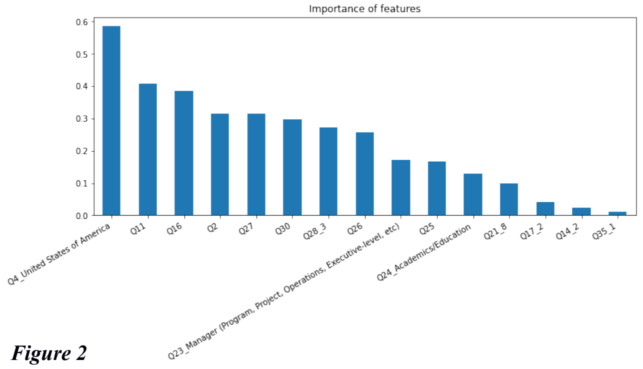
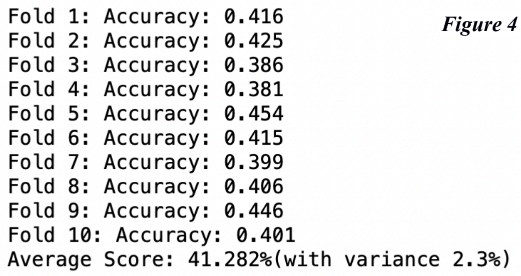


Figure 2

Part 3 Model Implementation

Ordinal logistic regression algorithm was implemented on the training data using 10-fold cross-validation. Pre-processing the reduced dataset by splitting it into 70% training and 30% testing dataset and scaling the input features separately, as both nominal 0 or 1 and larger-scale ordinal features were contained, was quite necessary before the model implementation, otherwise the features with greater scales would have a greater impact on the solution. For each iteration in each cross-validation algorithm, 14 binary logistic regression models were combined and the predictions of the salary bucket were determined by the highest probability of belonging. Then the accuracy were computed accordingly for the 10 folds. As a result indicated in *Figure 4*, all the 10 accuracies in predicting the salary bucket were not much high, with an avenge of 41.282%, and the variance among them was 2.3%. Here, in order to find the proper balance between the bias and variance for the imbalanced data, 9 models with different hyper-parameter c controlling the regularization strength inversely were built and compared. As the predictions were based on the highest probability of belonging, the mean average error between the probability of true prediction and our prediction was considered as the metric, for both training and validation dataset. With the resulted best c=0.01, the average accuracy dropped a little to 41.177% but with a smaller variance 2.0% .



```
Fold 1: Accuracy: 0.416
Fold 2: Accuracy: 0.425
Fold 3: Accuracy: 0.386
Fold 4: Accuracy: 0.381
Fold 5: Accuracy: 0.454
Fold 6: Accuracy: 0.415
Fold 7: Accuracy: 0.399
Fold 8: Accuracy: 0.406
Fold 9: Accuracy: 0.446
Fold 10: Accuracy: 0.401
Average Score: 41.282%(with variance 2.3%)
```

Figure 4

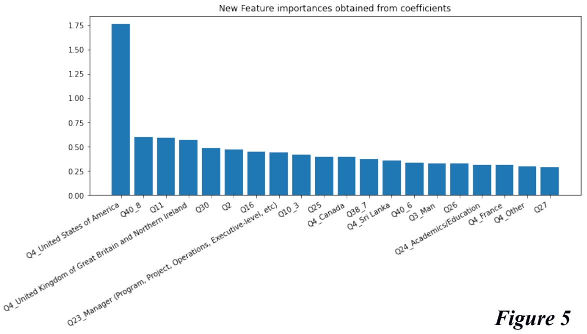


Figure 5

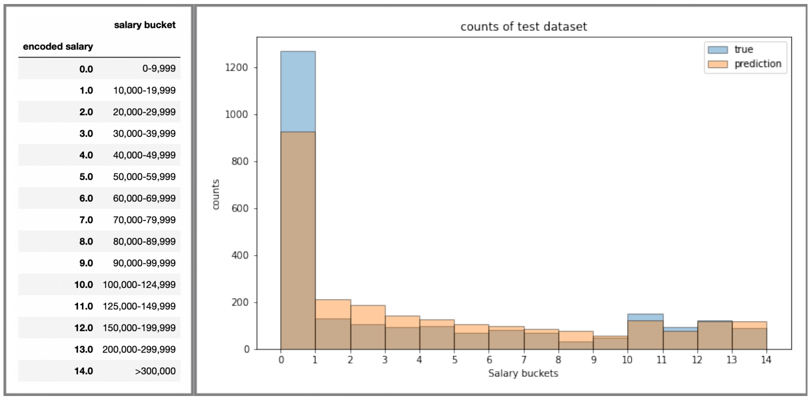
Part 4: Model tuning

For a logistic regression, the hyper-parameters can be tuned was ‘penalty’, ‘c', ‘solver’, which respectively shrinks the coefficients of the less contributive variables toward zero, controls the penalty strength, and represents the algorithm to use in the optimization problem. For this part, c and penalty was tuned with trying [0.001, 0.01, 0.05, 0.5, 1.0,10.0, 100.0] and ['newton-cg', 'lbfgs', 'liblinear', 'sag', ‘saga'] respectively. As a widely used, accuracy was not a suitable performance metric for this imbalance dataset as the salary bucket 0.0, which represented ‘0-9,999’ took the majority, so the model might fail to identify the minority class well for a better accuracy. So we could find a balance between precision and recall using F1-score and refit accuracy meanwhile. As a consequence, the final optimal model took c=10 and solver =‘lbfgs’, leading the best accuracy to 42.967%. With the best model, the 20 most deterring features were displayed in *Figure 5*, with many of them were picked previously, for example, residing in US of America was still the most determine feature on the salary, the age, being a manager, the years on writing code/programming or using machine learning methods also played an importance role as indicated before. There were also other features coming to the fore, like one of the option of Q40, whether Aporia was use to help monitor the machine learning model.

Part 5: Testing & Discussion

Predicted the testing data using the model above, got the F1 score of 0.363 and the accuracy of 40.475%. Both of them were below the scores of training data, indicating the performance on testing data was not as good as training data.The model was underrating as only a small part were trained and can be improved with more features involved. Plotting the overall distribution of the predicted salary bucket and the true salary bucket, both skewed right, as the smallest salary bucket took the majority as mentioned above. However, someones with 0-9999 US-dollars salary was predicted with higher payment, but it was not sufficient to determine where they were predicted as by only the distribution plots.

One drawback of ordinal encoding was that machine learning algorithms might interpret the encoded categories with certain relationships, like age 22-24 encoded with 1 might be considered as twice good as 18-21 encoded with 0.



	salary bucket
encoded salary	
0.0	0-9,999
1.0	10,000-19,999
2.0	20,000-29,999
3.0	30,000-39,999
4.0	40,000-49,999
5.0	50,000-59,999
6.0	60,000-69,999
7.0	70,000-79,999
8.0	80,000-89,999
9.0	90,000-99,999
10.0	100,000-124,999
11.0	125,000-149,999
12.0	150,000-199,999
13.0	200,000-299,999
14.0	>300,000

Figure 6

Reference

- *What is feature engineering? definition and faqs*. What is Feature Engineering? Definition and FAQs | HEAVY.AI. (n.d.). Retrieved November 9, 2022, from https://www.heavy.ai/technical-glossary/feature-engineering
- *What is feature engineering? definition and faqs*. What is Feature Engineering? Definition and FAQs | HEAVY.AI. (n.d.). Retrieved November 9, 2022, from https://www.heavy.ai/technical-glossary/feature-engineering
- Kassambara, mani3, Visitor, Francis, Don, & Kassambara. (2018, March 11). *Penalized logistic regression essentials in R: Ridge, Lasso and elastic net*. STHDA. Retrieved November 9, 2022, from http://sthda.com/english/articles/36-classification-methods-essentials/149-penalized-logistic-regression-essentials-in-r-ridge-lasso-and-elastic-net/